

Application/Control Number: 09/811,994  
Art Unit: 2157

Docket No.: PALM-3615

### **REMARKS**

Reconsideration and allowance are respectfully requested in view of the foregoing amendments and the following remarks.

By this amendment, claims 1-23 are pending. Applicants amended claim 1 and added new claims 21-23.

#### **Rejection of Claims 1, 6, 7, 13, 19 and 20**

On page 2 of the Final Office Action, the Examiner rejected claims 1, 6, 7, 13, 19 and 20 under 35 U.S.C. 102(e) as allegedly being anticipated by U.S. Patent No. 6,651,101 to Gai et al. ("Gai"). Applicants respectfully traverse the rejection. Applicants amended claim 1 only to improve presentation and not to overcome any prior art rejection. Applicants submit that the scope of claim 1 remains unchanged.

Independent claim 1 is directed to a communication protocol that includes, among other things, a first utility program adding a token, a first category type identifier corresponding to a first data type, and a first data type identifier corresponding to the first data type, to data to form an information packet and then, transparently to a sending application, using the transport mechanism to transmit the information packet to a second computer system.

On page 3 of the Final Office Action of March 11, 2005, the Examiner alleged that Gai, at col. 7, line 65 through col. 8, line 14, discloses this feature. Applicants disagree.

Gai, at col. 7, line 65 through col. 8, line 14, discloses:

In particular, upon initialization at host/server 222, the application program 224 preferably issues a StartUp( ) API call 410 to the API layer 236 at flow declaration component 226. Program 226 [sic] preferably loads the StartUp( ) call 410 with an application identifier that uniquely identifies application program 224 to component 226 as an argument. The application identifier may be a globally unique identifier (GUID), which is a 128 bit long value typically provided by the application developer, although other identifiers may also be used (e.g., application name). The StartUp( ) call 410 may be returned by the flow declaration component 226 with a version number as an argument.

Application/Control Number: 09/811,994  
Art Unit: 2157

Docket No.: PALM-3615

The version number corresponds to the version of software being executed by the flow declaration component 226. Other arguments, such as the quality-of-service (QoS) and/or traffic management resources that are available to traffic flows originating from program 224, may also be returned by flow declaration component 226.

Thus, Gai discloses an application program issuing a StartUp call via an application program layer (APL) at a flow declaration component. The application program loads the StartUp call with an application identifier that uniquely identifies the application program to the flow control component. The StartUp call may return a software version number, as well as other arguments to the application program (see Fig. 4A).

Applicants submit that the StartUp Call is made by the application program to introduce the application program to the flow declaration component and for the flow control component to provide to the application program its software version, as well as other parameters. The cited portion of Gai has absolutely nothing to do with adding a token, a first category type identifier, and a first data type identifier to data to form an information packet. Nothing in the cited portion of Gai, or anywhere else in Gai, discloses or suggests a first utility program adding a token, a first category type identifier corresponding to a first data type, and a first data type identifier corresponding to the first data type, to data to form an information packet and then, transparently to a sending application, using the transport mechanism to transmit the information packet to a second computer system, as required by claim 1.

Claim 1 further recites, among other things, a second utility program, resident on a second computer system, locating the first data type identifier and the first category type identifier in the information packet using said token. On page 3 of the Final Office Action, the Examiner alleged that Gai, at col. 16, lines 21-47, discloses this feature. Applicants disagree.

Gai, at col. 16, lines 21-47, discloses:

Application/Control Number: 09/811,994  
Art Unit: 2157

Docket No.: PALM-3615

As packets or frames are received at the local policy enforcer 210, they are examined by the packet/frame classifier 314. More specifically, the packet/frame classifier 314 parses the source and destination port fields 152, 154 (FIG. 1C) and the IP source and destination address fields 126, 128 and the protocol field 124 (FIG. 1B). This information is then supplied to the traffic flow state machine engine 310, which determines whether a traffic flow state has been established for such packets or frames. Assuming the packets or frames correspond to the anticipated flow from the program 224 to end station 212 (e.g., the IBM stock quote form), a traffic flow state will exist and have associated policy rules or service treatments as specified in the Policy Decision message 430 from policy server 216. Local policy enforcer 210 then applies the specified treatments to these packets or frames. For example, the traffic flow state machine engine 310 may instruct the packet/frame classifier, to set the DS field 132 (FIG. 1B) of such packets or frames to a value associated with best effort traffic. Similarly, the traffic flow state machine engine 310 may instruct the queue selector/mapping entity 318 to place these packets or frames in a particular (e.g., moderate priority) queue. Alternatively or in addition, packet/frame classifier may be instructed to load the ToS field 122 (FIG. 1B) or the user priority field 108 (FIG. 1A) with predetermined values so as to implement these treatments at other intermediate network devices, such as device 208.

Thus, Gai discloses a packet/frame classifier, included within a policy enforcer, examining received packets or frames. Source and destination port fields, source and destination IP address fields, and a protocol field are parsed by the packet/frame classifier. The traffic flow state machine uses this information to determine whether a traffic flow state exists for the frames or packets. If a traffic flow state exists, then the traffic flow will have associated policy rules or service treatments, which will be applied by local policy enforcer to the frames or packets.

Applicants submit that the cited portion of Gai, or any other portion of Gai fails to disclose or suggest a second utility program, resident on a second computer system, locating the first data type identifier and the first category type identifier in the information packet using the token, as required by claim 1. The source and destination port fields and the source and destination IP address fields are not the equivalent of a first data type identifier or a first category type identifier. Assuming *arguendo* that the protocol field includes a token, a point which Applicants do not concede, Gai fails to disclose or suggest using the token to locate the first data type identifier and the first category type identifier in the information packet.

Application/Control Number: 09/811,994  
Art Unit: 2157

Docket No.: PALM-3615

Therefore, Applicants submit that Gai fails to disclose or suggest a second utility program, resident on a second computer system, locating a first data type identifier and a first category type identifier in the information packet using a token, as required by claim 1.

On page 2 of the Advisory Action, dated July 5, 2005, the Examiner stated that he

interprets the first category identifier as a transaction type which Gai discloses as data information regarding user name, user department print and the first identifier as a subtransaction such as a print job on a HP laser jet printer.

The Examiner relied on Gai, col. 10, lines 8-25 for support. Applicants respectfully disagree with the Examiner

Gai, at col. 10, lines 8-27, states:

The application-level parameters may encompass a whole range of information relating to different aspects of the traffic flow from the application program 224. For example, application-level parameters include such information as user name (e.g., John Smith), user department (e.g., engineering, accounting, marketing, etc.), application name (e.g., SAP R/3, PeopleSoft, etc.), application module (e.g., SAP R/3 accounting form, SAP R/3 order entry form, etc.), transaction type (e.g., print), sub-transaction type (e.g., print on HP Laser Jet Printer), transaction name (e.g., print monthly sales report), sub-transaction name (e.g., print monthly sales report on A4 paper), application state (e.g., normal mode, critical mode, primary mode, back-up mode, etc.). For a video streaming application, the application-level parameters might include user name, film name, film compression method, film priority, optimal bandwidth, etc. Similarly, for a voice over IP application, the application-level parameters may include calling party, called party, compression method, service level of calling party (e.g., gold, silver, bronze), etc.

Thus, Gai discloses that data information regarding user name, user department, a subtransaction such as a print job on a HP laser jet printer are application-level parameters.

According to Gai, at col. 9, line 30 through col. 10, line 7, application-level parameters may be set by the application program issuing one or more "Set" API calls to the flow declaration component to cause the flow declaration component to load a traffic flow data structure with application-level parameters (also see Gai, at col. 8, lines 33-64 and Fig. 4A). "Get" API calls to the flow declaration component may retrieve data parameters stored in a traffic flow data structure (see Gai, at col. 10, lines 38-40 and Fig. 4B). However, the

Application/Control Number: 09/811,994  
Art Unit: 2157

Docket No.: PALM-3615

traffic flow data structure is not an information packet formed by adding a token, a first category type identifier and a first data type identifier to data, as required by claim 1. For example, Gai does not disclose or suggest using a transport mechanism to transmit the traffic flow data structure to a second computer system, which would be required by claim 1 if the traffic flow data structure were considered an information packet. Thus, assuming *arguendo*, that the application-level parameters include a first category identifier and a first data type identifier, a point which Applicants do not concede, Gai is completely silent regarding any disclosure or suggestion of a first utility program adding a token, a first category type identifier corresponding to the first data type, and a first data type identifier corresponding to the first data type, to the data to form an information packet and then, transparently to a sending application, using a transport mechanism to transmit the information packet to a second computer system, as required by claim 1.

Because Gai fails to disclose or suggest each and every feature of claim 1, Applicants submit that claim 1 and dependent claims 6 and 7 are not anticipated by Gai. Applicants, therefore, respectfully request that the rejection of claims 1, 6 and 7 be withdrawn.

Claim 13 recites features similar to those of claim 1 and is not anticipated by Gai for at least reasons similar to those discussed with respect to claim 1. Therefore, Applicants respectfully request that the rejection of claim 13 and dependent claims 19 and 20 be withdrawn.

#### **Rejection of Claims 2-5, 8-12 and 14-18**

On page 4 of the Final Office Action, the Examiner rejected claims 2-5, 8-12 and 14-18 under 35 U.S.C. 103(a) as allegedly being unpatentable over Gai and further in view of U.S. Patent No. 6,654,786 to Fox et al. ("Fox"). Applicants respectfully traverse the rejection.